05.14.04

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Oleg Kiselev		
Assignee:	VERITAS Software Corporation		
Title:	Application-Assisted Recovery from Data Corruption in Parity RAID Storage Using Successive Re-reads		
Application No.:	10/614,306	Filing Date:	July 3, 2003
Examiner:	Unassigned	Group Art Unit:	2186
Docket No.:	**VRT0059US**	Confirmation No.:	8239

Austin, Texas
May 14, 2004

Mail Stop Petition
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313-1450

PETITION TO MAKE SPECIAL UNDER 37 CFR §1.102(d)

Dear Sir:

The applicants hereby petition pursuant to 37 CFR §1.102(d) and MPEP § 708.02(VIII) to make the above-identified application special. Please charge Deposit Account No. 502306 for the fee of $130.00 for this petition as set forth in 37 CFR §1.17(h).

Should the Office determine that all the claims presented are not obviously directed to a single invention, the applicants will make an election without traverse as a prerequisite to the grant of special status.

05/17/2004 [illegible] 00000135 502306 10614306
01 FC:1460 130.00 DA

The applicants respectfully submit that a pre-examination search has been performed by a professional search firm in the following classes/subclasses:

Class	Subclasses
G06F (Electrical Digital Data Processing)	03/06 12/08 12/10 12/16 13/14
714 (Error Detection/Correction and Fault Detection/Recovery)	006 007 048
711 (Electrical Computers and Digital Processing Systems: Memory)	004 136 114 154 202

Enclosed are copies of the following references which are presently believed to be, from among those made of record in the accompanying Information Disclosure Statement and any previously filed Information Disclosure Statement, the most closely related to the subject matter encompassed by the claims:

Japanese Patent No. 11-224166	Kuniaki
U.S. Patent No. 6,327,638 B1	Kirby
U.S. Patent No. 6,460,122 B1	Otterness et al.
U.S. Patent No. 6,704,837 B2	Beardsley et al.

Detailed Discussion of the References

U.S. Patent No. 6,460,122 issued October 1, 2002, to Otterness et al. ("Otterness"). Otterness provides a multiple level cache structure and multiple level caching method that distributes I/O processing loads including caching operations between processors to provide high performance I/O processing, especially in a server environment. Otterness achieves optimal data throughput by taking advantage of multiple processing resources. Otterness manages the allocation of the data caches to optimize the host access time and parity generation.

Otterness teaches a cache allocation for RAID stripes guaranteed to provide fast access times for the XOR engine by ensuring that all cache lines are allocated from the same cache level. Otterness discloses allocation of cache lines for RAID levels which do not require parity generation and are allocated in such manner as to maximize use of the memory bandwidth. Parity generation which is optimized for use of the processor least utilized at the time the cache lines are allocated, thereby providing for dynamic load balancing amongst the multiple processing resources, is disclosed. A cache line descriptor for maintaining information about which cache data pulled the cache line resides within, and a cache line descriptor which includes enhancements to allow for movement of cache data from one cache level to another is disclosed. The cache line descriptor with enhancements for tracking the cache within which RAID strip cache lines siblings reside is disclosed.

Claim 1 of the instant application requires (1) receiving a first request to read data of a stripe unit B_x of a stripe; (2) returning data of stripe unit B_x; (3) receiving a second request to read data of stripe unit B_x; (4) generating new data for stripe unit B_x in response to receiving the second request; and (5) returning the new data. While Otterness describes use of a RAID storage system, the foregoing description of Otterness does not teach nor fairly suggest the limitations (1) – (5) either alone or in combination with the remaining limitations of claim 1. Accordingly, Applicant submits that the independent claim 1 is distinguishable over Otterness.

Claim 15 of the instant application requires (A) comparing an identification for stripe unit B_x with identifications for stripe units stored in a table in memory; (B) if the identification for stripe unit B_x does not compare equally then storing the identification for stripe unit B_x in the table and returning data of the stripe unit; (C) if the identification for stripe unit B_x does compare equally then generating new data for stripe unit B_x and returning the new data. Otterness does not teach nor fairly suggest the limitations (A) – (C) either alone or in combination with the remaining limitations of claim 15. Accordingly, Applicant submits that the independent claim 15 is distinguishable over Otterness.

U.S. Patent 6,327,638 issued December 4, 2001, to Kirby ("Kirby"). Kirby relates to a disk striping method and storage subsystem using the disk striping method. Kirby uses the method to assure substantially a constant performance across all zones of the disk drive when transferring data to or from the disk drive. Generally, Kirby stripes data over a plurality of disks such that half the disks map sequential block addresses from outer to inner zones of the disk drive and the other half maps addresses from the inner to outer zones. For example, where half the data in a stripe is manipulated on faster outer zones, the other half is manipulated on correspondingly slower inner zones of the disk drive. Or where half the data in a stripe is manipulated on middle zones, the other half of the data is also manipulated in middle zones thereby achieving a more consistent average sustained I/O performance.

While Kirby describes a RAID subsystem for storing data, Kirby fails to teach or fairly suggest (1) receiving a first request to read data of a stripe unit B_x of a stripe; (2) returning data of stripe unit B_x; (3) receiving a second request to read data of stripe unit B_x; (4) generating new data for stripe unit B_x in response to receiving the second request; and (5) returning the new data, either alone or in combination with the remaining limitations of claim 1. Moreover, Kirby fails to teach or fairly suggest (A) comparing an identification for stripe unit B_x with identifications for stripe units stored in a table in memory; (B) if the identification for stripe unit B_x does not compare equally then storing the identification for stripe unit B_x in the table and returning data of the stripe unit; (C) if the identification for stripe unit B_x does compare equally then generating new data for

stripe unit B_x and returning the new data, either alone or in combination with the remaining limitations of claim 15.

U.S. Patent No. 6,704,837 to Beardsley et al. ("Beardsley") issued March 9, 2004. Beardsley relates to a method and apparatus for increasing RAID write performance by maintaining a full track write counter. Figure 1 illustrates a block diagram of RAID system 100 employing Beardsley's invention. System 100 includes a RAID array of hard disks 146 – 152. In the past, when performing a RAID write, the RAID write may not contain a stripe with a full tracks. To perform the write, the parity must first be read, then the new parity generated, and finally the data and new parity can be written. To optimize this process, the parity read can be avoided by writing a stripe width of full tracks. During a write, an assumption that a stripe of full tracks exist must be made and then the tracks are grouped. Nevertheless, during the grouping, the controller may discover that a stripe of full tracks does not exist, yet the write will still include a parity read and the overhead doing the track grouping has been incurred.

Beardsley purports to avoid unnecessary track grouping during writes by using a full track write counter. When a write request is received in Beardsley, a full track write counter for tracks and a stripe of tracks associated with the write request is analyzed to determine whether the write request involves a full track write. Beardsley then describes subsequently executing a cache destage based on the analysis of the full track write counter for tracks and a stripe of tracks associated with the write request.

While Beardsley describes use of a RAID storage subsystem, the foregoing description of Beardsley fails to teach or fairly suggest (1) receiving a first request to read data of a stripe unit B_x of a stripe; (2) returning data of stripe unit B_x; (3) receiving a second request to read data of stripe unit B_x; (4) generating new data for stripe unit B_x in response to receiving the second request; and (5) returning the new data, either alone or in combination with the remaining limitations of claim 1. Moreover, Beardsley fails to teach or fairly suggest (A) comparing an identification for stripe unit B_x with identifications for stripe units stored in a table in memory; (B) if the identification for stripe unit B_x does not compare equally then storing the identification for stripe unit B_x in the table and returning data of the stripe unit; (C) if the identification for stripe unit B_x

does compare equally then generating new data for stripe unit B_x and returning the new data, either alone or in combination with the remaining limitations of claim 15.

Japanese Patent 11-224166 issued to Kuniaki ("Kuniaki") based upon application No. 10024592. Kuniaki relates to a fault avoiding method for storage area in log in type disk storage device and computer readable storage medium used in the same device. According to the abstract of 10024592, Kuniaki discloses a system which includes a logical address conversion table 22, a stripe management table 23, a redundant information generating means 24, a data recovering means 25, an access limitation start fixed fault stripe number storage means 26, and write buffer 27. Kuniaki can improve avoidance of a fault due to a fixed failure in a heart of a storage device.

While Kuniaki appears to employ RAID storage, the English abstract of Kuniaki fails to teach or fairly suggest (1) receiving a first request to read data of a stripe unit B_x of a stripe; (2) returning data of stripe unit B_x; (3) receiving a second request to read data of stripe unit B_x; (4) generating new data for stripe unit B_x in response to receiving the second request; and (5) returning the new data, either alone or in combination with the remaining limitations of claim 1. Moreover, the English abstract fails to teach or fairly suggest (A) comparing an identification for stripe unit B_x with identifications for stripe units stored in a table in memory; (B) if the identification for stripe unit B_x does not compare equally then storing the identification for stripe unit B_x in the table and returning data of the stripe unit; (C) if the identification for stripe unit B_x does compare equally then generating new data for stripe unit B_x and returning the new data, either alone or in combination with the remaining limitations of claim 15.

CONCLUSION

Applicant respectfully requests that this petition be granted, and that the present application receive expedited examination. Should any issues remain that might be subject to resolution through a telephonic interview, the Office is requested to telephone the undersigned.

EXPRESS MAIL NO:
EV 304739210 US

Respectfully submitted,

Eric A. Stephenson
Attorney for Applicant(s)
Reg. No. 38,321
512-439-5093 (Phone)
512-439-5099 (Fax)